

Amendments to the Claims

1. (Currently Amended) A method for shared cache invalidation, comprising:

hashing a key corresponding to an object in a first cache that has been modified or deleted to provide a hashed code of the key, wherein the first cache forms part of a shared cache;

transmitting only the hashed code of the key to other caches in the shared cache;

comparing the hashed code of the key with entries in the other caches;
and

dropping any keys in the other caches having a hash code the same as the hashed code of the key.

2. (Original) The method of claim 1, further comprising:

receiving at the first cache a hashed code of a key from one of the other caches in the shared cache corresponding to an object that has been modified or deleted;

comparing the received hashed code of the key with entries in the first cache; and

dropping any keys in the first cache having a hash code the same as the received hashed code of the key.

3. (Original) The method of claim 1, wherein entries in the other caches of the shared cache are indexed using hash codes.

4. (Currently Amended) A method for shared cache invalidation, comprising:

hashing a key corresponding to an object in a cache that has been modified or deleted to provide a hashed code of the key, wherein the cache forms part of a shared cache; and

transmitting only the hashed code of the key to other caches in the shared cache for invalidation processing.

5. (Original) The method of claim 4, wherein the invalidation processing further comprises:

comparing the received hashed code of the key with entries in the other caches; and

dropping any keys in the other caches having a hash code the same as the hashed code of the key.

6. (Currently Amended) ~~The method of claim 4, further~~ A method for shared cache invalidation, comprising:

hashing a key corresponding to an object in a cache that has been modified or deleted to provide a hashed code of the key, wherein the cache forms part of a shared cache;

transmitting the key and the hashed code of the key to the other caches in the shared cache for invalidation processing;

at each of the other caches in the shared cache:

hashing the key to provide a second hashed code of the key; and

comparing the hashed code of the key and the second hashed code of the key;

wherein the method for shared cache invalidation is not applicable to the key if the hashed code of the key and the second hashed code of the key are not the same at any of the other caches.

7. (Currently Amended) A method for shared cache invalidation, comprising:

receiving at a cache a key and a hashed code of a the key, wherein the hashed code of the key corresponds to an object that has been modified or deleted in another cache; ~~and~~

performing invalidation processing on the cache based on the received hashed code of the key;

hashing the key to provide a second hashed code of the key; and

comparing the hashed code of the key and the second hashed code of the key;

wherein the invalidation processing is not performed if the hashed code of the key and the second hashed code of the key are not the same at any of the other caches.

8. (Original) The method of claim 7, wherein the invalidation processing further comprises:

comparing the received hashed code of the key with entries in the cache;
and

dropping any keys in the cache having a hash code the same as the received hashed code of the key.

9. (Currently Amended) A system for shared cache invalidation, comprising:

a system for hashing a key corresponding to an object in a cache that has been modified or deleted to provide a hashed code of the key, wherein the cache forms part of a shared cache; and

a system for transmitting only the hashed code of the key to other caches in the shared cache for invalidation processing.

10. (Original) The system of claim 9, further comprising a system for providing the invalidation processing at the other caches, wherein the system for providing the invalidation processing comprises:

a system for comparing the hashed code of the key with entries in the other caches; and

a system for dropping any keys in the other caches having a hash code the same as the hashed code of the key.

11. (Currently Amended) A system for shared cache invalidation, comprising:

a system for receiving at a cache a key and a hashed code of a the key, wherein the hashed code of the key corresponds to an object that has been modified or deleted in another cache; ~~and~~

a system for performing invalidation processing on the cache based on the received hashed code of the key;

a system for hashing the key to provide a second hashed code of the key;
and

a system for comparing the hashed code of the key and the second hashed code of the key;

wherein the invalidation processing is not performed if the hashed code of the key and the second hashed code of the key are not the same at any of the other caches.

12. (Original) The system of claim 11, wherein the system for performing invalidation processing further comprises:

a system for comparing the received hashed code of the key with entries in the cache; and

a system for dropping any keys in the cache having a hash code the same as the received hashed code of the key.

13. (Currently Amended) A program product for shared cache invalidation stored on a recordable medium, which when executed, comprises:

program code for hashing a key corresponding to an object in a cache that has been modified or deleted to provide a hashed code of the key, wherein the cache forms part of a shared cache; and

program code for transmitting only the hashed code of the key to other caches in the shared cache for invalidation processing.

14. (Original) The program product of claim 13, further comprising program code for invalidation processing, the program code for invalidation processing comprising:

program code for comparing the received hashed code of the key with entries in the other caches; and

program code for dropping any keys in the other caches having a hash code the same as the hashed code of the key.

15. (Currently Amended) A program product for shared cache invalidation stored on a recordable medium, which when executed, comprises:

program code for receiving at a cache a key and a hashed code of a the key, wherein the hashed code of the key corresponds to an object that has been modified or deleted in another cache; ~~and~~

program code for performing invalidation processing on the cache based on the received hashed code of the key;

program code for hashing the key to provide a second hashed code of the key; and

program code for comparing the hashed code of the key and the second hashed code of the key;

wherein the invalidation processing is not performed if the hashed code of the key and the second hashed code of the key are not the same at any of the other caches.

16. (Original) The program product of claim 14, wherein the program code for performing invalidation processing further comprises:

program code for comparing the received hashed code of the key with entries in the cache; and

program code for dropping any keys in the cache having a hash code the same as the received hashed code of the key.